

RCA PF0220096 RefAA

CITED BY APPLICANT

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



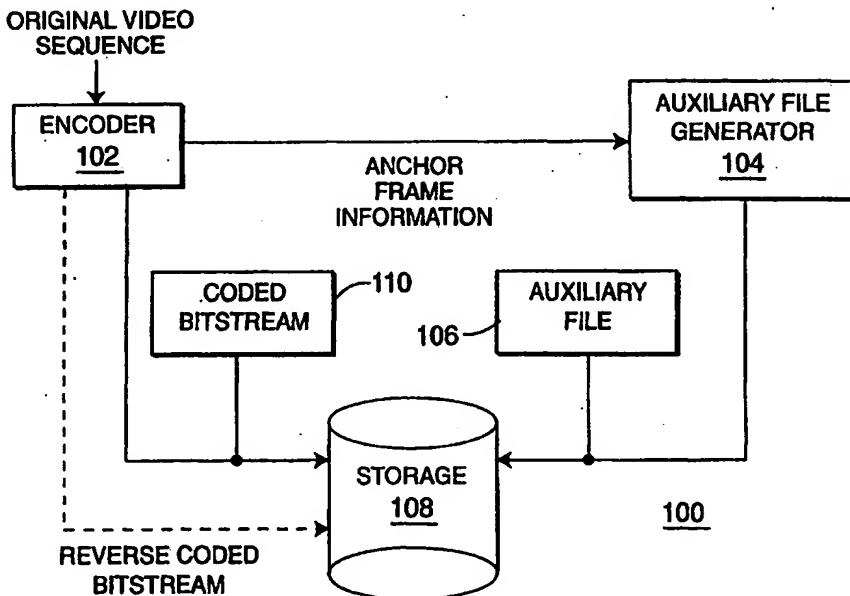
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ :	A1	(11) International Publication Number: WO 00/22820
H04N 5/783		(43) International Publication Date: 20 April 2000 (20.04.00)
(21) International Application Number: PCT/US99/23375		(81) Designated States: BR, CA, CN, IN, JP, KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 7 October 1999 (07.10.99)		
(30) Priority Data: 60/103,762 9 October 1998 (09.10.98) US		Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>
(71) Applicant: SARNOFF CORPORATION [US/US]; 201 Washington Road, CN 5300, Princeton, NJ 08543-5300 (US).		
(72) Inventors: PEJHAN, Sasan; 4 Heritage Boulevard, Princeton, NJ 08540 (US). FESTA, John; 206 Central Avenue, Metuchen, NJ 08840 (US).		
(74) Agents: NEY, Andrew, L.; Ratner & Prestia, 301 One Westlakes (Berwyn), P.O. Box 980, Valley Forge, PA 19482-0980 (US) et al.		

(54) Title: METHOD AND APPARATUS FOR PROVIDING VCR-TYPE CONTROLS FOR COMPRESSED DIGITAL VIDEO SEQUENCES

(57) Abstract

A method and apparatus that provides dynamic frame rate control for compressed digital video sequences to facilitate applying VCR-like controls to the compressed digital video sequences. More specifically, for each compressed video file produced by an encoder (102), there is an associated auxiliary file, e.g., with the same prefixes as the compressed file but with a "VCR" suffix, that would contain information about the intracoded frames (I-frames) of the compressed bitstream. This auxiliary file is created at the time that the original raw video sequence is being encoded (compressed), and contains information about the position of I-frames within the compressed bitstream. These I-frames, which can be independently decoded by a decoder (504/604), serve as reference points for decoding



and displaying other predicted frames. Thus, to perform random frame access, for example, the decoder (504/604) can find the position of the nearest I-frame preceding the frame of interest by accessing the auxiliary file. The system can then decode the I-frame without displaying that decoded frame and then decode and display the frame of interest. This technique can randomly access any frame in a compressed sequence to implement fast forward and fast reverse functions.

THIS PAGE BLANK (USPTO)

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Republic of Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

-1-

METHOD AND APPARATUS FOR PROVIDING VCR-TYPE CONTROLS FOR COMPRESSED DIGITAL VIDEO SEQUENCES

This application claims benefit of U.S. provisional patent application serial number 60/103,762, filed October 9, 1998 and herein incorporated by reference.

5 The invention generally relates to digital multimedia communication systems and, more particularly, to methods and apparatus for providing VCR type controls for compressed digital video sequences.

BACKGROUND OF THE INVENTION

As a result of the wide-spread use of powerful and multimedia friendly personal 10 computers, it has become increasingly desirable to generate and view digital video clips. Video clips are becoming abundant on many INTERNET web sites and have been available on CD-ROMs for many years now. Unlike other traditional media, such as audio and text, video clips in their raw format can become prohibitively large computer files, consuming storage and bandwidth at unacceptably high rates. A substantial amount of 15 research has therefore been performed over the past 30 years to develop efficient video compression algorithms. Several standards, including MPEG (-1, -2, -4), H.261 and H.263 have been developed. Almost all digital video sequences, whether on the web, on CD-ROMs or on local hard disks, are stored in one compressed format or another.

Given the ease with which video clips can be played back on a computer, there is a 20 natural desire to have the same type of controls as one has with regular (analog) video players (such as Play, Stop/Pause, Fast Forward, Slow Motion, Reverse) as well as more sophisticated controls, such as random frame access, jumping to the beginning or end of a clip, or jumping to the next or previous scene. Such controls can easily be implemented for raw video: the size of the frames are fixed, the position of each frame in the bitstream 25 is known and frames can be accessed and displayed independently from one another. For compressed video, implementing some of these controls is challenging. Compressed frames have a variable size and their position in the bitstream may not be readily available. Moreover, if predictive coding is used (such as motion compensation), a given frame may not be decodable independently of other frames in the sequence. The 30 operations which are simple to implement for compressed streams include Play,

-2-

Stop/Pause, Slow Motion and Rewind, which are currently performed by most standard software decoders. The challenging operations include random frame access, fast forward, playing in reverse and jumping to the next scene change.

Brute force solutions to these challenges could be implemented using a very powerful computer, or when dealing with very small resolution and/or relatively short sequences. For instance, the Fast Forward control could be implemented by decoding and displaying the clip at two or three times the natural speed. With high resolutions and long sequences, however, this is not a practical option, particularly in cases where the video is being streamed over a network.

Therefore, there is a need in the art for a method and apparatus of providing VCR-type controls to a compressed video bitstream.

SUMMARY OF THE INVENTION

The disadvantages associated with the prior art are overcome by a method and apparatus for efficient implementation of VCR-type controls to manipulate a compressed video clip. More specifically, for each compressed video file, there is an associated auxiliary file (e.g., with the same prefix as the file name that identifies the compressed video clip, but with a 'vcr' suffix). The VCR auxiliary file is generated during the encoding process and contains information pertaining to the position of anchor frames in the compressed bitstream. Using this information, VCR-type controls can be implemented for both local and remote files. The use of the VCR auxiliary file enables a computer to utilize VCR-type controls such as random frame access, jumps to next/previous scene, fast forward and reverse. To perform random frame access, for example, the video decoder finds the position of the nearest anchor frame preceding the frame of interest by searching the auxiliary file. The decoder then proceeds to decode that frame and subsequent frames (without displaying the decoded frames, though) until the decoder reaches the frame of interest, which is both decoded and displayed. Both Fast Forward and Reverse play are implemented as special cases of Random Frame Access, i.e., a sequence of intermittently decoded frames are displayed in forward or reverse order to produce fast forward or fast reverse control.

-3-

An alternative implementation that performs reverse play, caches a Group of Pictures (i.e., caches all the frames between to neighboring anchor frames). Then, the cached frames can be displayed at any speed in reverse order.

BRIEF DESCRIPTION OF THE DRAWINGS

5 The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

FIG. 1 depicts a block diagram of a video sequence encoder in accordance with the present invention that produces a compressed bitstream and an associated auxiliary file;

10 FIG. 2 depicts a file structure for a first embodiment of an auxiliary file;

FIG. 3 depicts a file structure for a second embodiment of an auxiliary file;

FIG. 4 depicts a file structure for a third embodiment of an auxiliary file;

FIG. 5 depicts a block diagram of a decoder for decoding bitstreams produced by the encoder of FIG. 1; and

15 FIG. 6 depicts a block diagram of a client and server for streaming, decoding, and displaying remote bitstreams produced by the encoder of FIG. 1.

To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION

20 The major video coding techniques/standards in use today include H.263, geared towards low bit-rate video, MPEG-1, developed for CD-ROM applications at around 1.5 Mbits/s and MPEG-2, designed for very high quality video (HDTV) at around 10 Mbits/s. Although there are major differences among these three standards, they are all based on the same basic principles described below.

25 Each frame of video can be one of three types: Intra-coded (I) frames (i.e., anchor frames), Predicted (P) frames and Bi-directionally predicted (B) frames. The I-frames are encoded very much like still images (i.e., JPEG) and achieve compression by reducing spatial redundancy: a Discrete Cosine Transform (DCT) operation is applied to 8x8 blocks

of pixels within the frame, starting from the top, left block and moving to the right and down the rows of pixels. To complete the encoding of an I-frame, the DCT coefficients are then quantized and entropy encoded.

The P-frames are predicted from a preceding I- or P-frame. Using motion estimation techniques, each 16x16 MacroBlock (MB) in a P-frame is matched to the closest MB of the frame from which it is to be predicted. The difference between the two MBs is then computed and encoded, along with the motion vectors. As such, both temporal and spatial redundancy is reduced. B-frames are coded in a manner similar to P-frames except that B-frames are predicted from both past and future I- or P-frames.

I-frames are much larger than P or B frames, but they have the advantage of being decodable independent of other frames. P and B frames achieve higher compression ratios, but they depend on the availability of other frames in order to be decoded.

The first embodiment of the invention for implementing VCR type controls generates a small, separate auxiliary 'vcr' file for each compressed video sequence (bitstream). This auxiliary file contains key information about the associated bitstream that enables efficient implementation of VCR type controls. Specifically, for each compressed video stream there is an associated auxiliary file (e.g., with the same prefix as the compressed file name but with a 'vcr' suffix). This auxiliary file primarily contains information about the position of Intra-coded frames (I-Frames) within the compressed bitstream.

Fig. 1 depicts a block diagram of a video sequence encoder system 100 containing an encoder 102, an auxiliary file generator 104 and a storage device 108 that operate in accordance with the present invention.

The encoder 102 encodes a video sequence in a conventional manner (e.g., as described above), but also produces I-frame information for use by the auxiliary file generator 104. This information pertains to the location of the I-frames within the encoded bitstream 110, e.g. the position of the I-Frame with respect to the beginning (or end) of the bitstream. The auxiliary file generator 104 produces an auxiliary file 106 for each encoded bitstream 110.

-5-

Video sequences may be encoded at either a variable or constant frame rate. The former may occur when encoders drop frames, in an irregular fashion, in order to achieve a constant bit-rate. Furthermore, even if the frame rate is constant, I-Frames may or may not occur at fixed intervals. Such aspects of the coding process are not specified by the standards but are left to implementers. For some applications, it may make sense to insert I-Frames at fixed intervals (e.g., every 30th frame can be an I-Frame). For other applications, implementers may decide to insert an I-Frame only whenever there is a scene change - something which may occur at irregular time intervals. The auxiliary file has a different format depending on whether I-Frames are inserted at fixed or variable intervals.

When the I-frames are contained at fixed intervals, i.e., the I-frames are generated at a fixed interval by the encoder 102, the auxiliary file 106 has a particular form that facilitates efficient implementation of the invention. FIG. 2 illustrates the format of an auxiliary file 200 for use with a bitstream having a fixed I-frame interval. The auxiliary file 200 contains a field 202, e.g., one byte, at the head of the file indicating the size of the fixed interval. Next, a field 204, e.g., four bytes for every I-frame is included in the header to indicate the offset from the beginning (or the end) of the bitstream at which each I-frame is located.

If the interval between I-Frames is variable, the auxiliary file 200 of FIG. 2 is augmented with additional information to become the auxiliary file 300 of FIG. 3. The first field 302 of auxiliary file 300 is still that of the I-frame interval, but the field value is set to 0 (or some other special code) to indicate a variable frame rate. There will now be 2 fields per I-Frame: Field 306 containing a 2-byte frame number and field 308 containing the 4-byte offset information. Field 304, which indicates the total number of frames in the entire sequence, can be optionally added to the auxiliary file 300 (placed right after the frame interval field 302). As will be described below, this optional information can help speed up the implementation of the random frame access control.

Finally, a one-bit Scene Change Indicator (SCI) field can be inserted in the auxiliary file for each I-Frame, indicating whether there has been a scene change or not from the previous I-Frame. One way of inserting this field is to add another one-byte field 310 for each I-frame, with the first bit serving as the SCI and the other bits reserved for

-6-

future use. Alternatively, the first bit of the 4-byte offset field 308 can be designated as the SCI field 312, with the remaining 31 bits used for the offset, as shown in FIG. 4.

Since the file format 300 for variable I-frame intervals is a superset of the one for fixed I-frame intervals (format 200), it could be used for both cases. This makes the 5 implementation of the invention slightly easier. The additional 2-bytes per I-frame will make the auxiliary files larger, however, for the case of fixed I-frame interval. Whether the trade-off is worth it or not is a choice for implementers to make and will vary from one case to another. All in all, however, the size of the auxiliary files generated is negligible compared to the size of the compressed video file. For the fixed I-frame 10 interval case, the size is basically four bytes multiplied by the number of I-frames. If I-frames are inserted as frequently as even three times a second (i.e. once every tenth frame), then the auxiliary file adds 12 bytes (84 bits) per second. Even for a very low bit-rate sequence (say 5 kbit/s) the additional storage required for the auxiliary file is negligible. For the case of variable I-frame interval, the size of the auxiliary file is 15 approximately six bytes multiplied by the number of I-frames. That translates into 126 bits/s, assuming three I-frames per second on the average.

FIG. 5 and FIG. 6 depict block diagrams of two different systems ("players") for playback of compressed video bitstreams with VCR-type controls. FIG. 5 depicts a player 500 that operates to playback locally stored video files. This player 500 comprises a User Interface/Display 502, a decoder 504, an auxiliary file processor 506 and local 20 storage 108. The user interacts with the system through the user interface (e.g., a graphical user interface that has various "buttons" for VCR controls). The decoded bitstream may also be displayed here or on a separate display. The decoder 504 operates like any standard decoder, except that when VCR commands are issued, it interacts with the auxiliary file processor 506 to determine the location in the bitstream from where the 25 decoder needs to start decoding. The auxiliary file processor 506 in turn retrieves that information from the auxiliary file. Both the bitstream and the associated auxiliary file are stored locally on the storage device 108.

FIG. 6 depicts a system 600 where the bitstream and associated auxiliary file are 30 stored remotely on a server 602. The bitstream is streamed to the player 601 over a

network 612. When the user Interface/Display 602 processes a VCR command issued by the user, the decoder 604 relays this command over the network 612 to the server 602. A buffer 610 located between the network 612 and the decoder 604. The server 602 then interacts with the auxiliary file processor 606, which now resides on the server 602, to determine the location within the bitstream from which the server should start transmission.

When a user (client) requests to view a random frame within a compressed video bitstream (i.e., the user requests random frame access) within either system 500 or 600, there are three cases to consider:

- 10 1. When the frame to be decoded (selected frame) is the one after the last decoded frame, this is the normal video sequence play mode for the systems 500 and 600. Thus, the decoder 504 or 604 operates in a conventional manner without needing to retrieve any information from the auxiliary file, i.e., the decoder sequentially selects frames for decoding and display.
- 15 2. When the frame to be decoded (selected frame) is sometime in the future relative to the frame that is presently being decoded, but before the next I-frame, the system 500 or 600 needs to decode frames as in the usual play mode using the interframe predictions but without displaying the decoded frames until the desired frame is reached. As such, the decoder 504/604 blocks display of the decoded frames until the selected frame is decoded.
- 20 3. All other cases require special handling. First the I-frame prior to the selected frame has to be identified, i.e., the I-frame prior to the selected frame must be identified when given the current frame number being decoded and given the fact that the first frame in the sequence is an I-frame. If the I-frame interval is fixed, the selected frame is easily determined. Next, the offset of the I-frame will be read from the auxiliary file 200 and provided to the decoder 504/604. Since there is a fixed size for the auxiliary file header and a fixed sized field (4-byte field 204) for each I-frame, determining the offset is trivial. The bitstream pointer that selects frames for decoding in the decoder 504/604 would then be moved according to the offset retrieved from the auxiliary file. The I-

-8-

frame and the subsequent P-frames would be decoded but not displayed until the selected frame is decoded.

When the compressed bitstream is stored remotely (as in FIG. 6), there will be no changes for scenarios one and two above since all frames have to be retrieved and decoded though not displayed sequentially. For scenario three, once the system 600 determines the last I-frame prior to the frame of interest, the system 600 sends a stop message to the server 602 supplying the bitstream, only by a request for the I-frame mentioned. The server 602 then looks up the offset of that I-frame in the auxiliary file at the server 602, reset the pointer to the compressed bitstream file, and start transmitting bits from that point. The rest of the decoding process remains as described above.

When a variable I-frame interval is used, as before, when a frame needs to be decoded, one of the three cases listed above would apply. For the first case (decoding the next frame) there is no difference. For the second and third cases, however, the decoder has to determine if there is an I-Frame which preceded the frame of interest or not. To this end, it has to look up the 2-byte frame numbers (field 306) in the auxiliary file 300 and extract the appropriate I-Frame accordingly.

To speed up the search for the appropriate I-Frame, the field 304 indicating the total number of frames in the entire sequence is used. As such, the server 602 compares the number of the frame to be decoded with the total number of frames in the sequence and determines an estimate of where in the auxiliary file the server wants to start the I-frame search.

The scenarios described above apply to both local and remote files. The only difference occurs in the third case: for local files, the player (client) 500 has to perform the tasks indicated; for remote files, the client 601 sends a request to a server 602 for a random frame, and the server 602 has to look up the auxiliary file and resume transmission from the appropriate place in the bitstream.

When a user requests a jump to the next or previous scene, the auxiliary file will be scanned to find the next/previous I-Frame that has the scene change bit set to TRUE. That frame is decoded and displayed and the clip starts playing from that point onwards. Algorithms for detecting scene changes are well-known in the art. An algorithm that is

-9-

representative of the state of the art is disclosed in Shen et al., "A Fast Algorithm for Video Parsing Using MPEG Compressed Sequences", International Conference on Image Processing, Vol. 2, pp. 252-255, October 1995.

5 In the present invention the auxiliary file information is used to provide a fast forward effect in the decoded video. The Fast Forward operation can simply be viewed as a special case of random access. Running a video clip at, say, three times its natural speed by skipping two out of every three frames is equivalent to continuously making 'random' requests for every third frame (i.e., requesting frame 0, 3, 6, 9, and so on). Every time a frame is requested, the random frame access operation described above first 10 determines the position of the nearest preceding I-frame just as before. The frames from that I-frame to the selected frame are decoded but not displayed. As such, only the requested frames are displayed, i.e., every third frame.

15 In addition to frame jumping and fast forward, the invention includes two embodiments for implementing Reverse play (both normal and fast speed). The first embodiment, which is simpler but less efficient, is to view the reverse play as a special case of random frame access. For each frame, the server or local decoder invokes the random frame access mechanism described above. The number of the 'random' frame to be retrieved is decremented by one or N each time, depending on the playback speed. This scheme is inefficient due to the existence of predicted frames. To see why, consider 20 the following case:

25 Assume a sequence where every 10th frame (0,10,20, and so on) is an I-Frame, and all other frames are forward predicted (P-) frames. To play this video clip in reverse, using the random access method, some of the frames have to be decoded several times. For example, after frame 10 is decoded and displayed, frames 0 through 8 are decoded (but not displayed) so that frame 9 can be decoded and displayed. Then, frames 0 through 8 are decoded again so that frame 8 can be displayed, and so forth. As such, the same frames must be decoded over and over to achieve a sequence of frames to be displayed in reverse order. The need for repetitively decoding the same frames is very inefficient.

30 The second embodiment involves caching in memory (cache 508 or 608) all the frames in a Group of Pictures (GOP) when the Reverse control is invoked. Thus, in the

-10-

example above, after frame 10 (an I-Frame) has been decoded and displayed, the decoder 504/604 can decode all frames between 0 and 9, cache them in memory (508/608 in FIGS. 5 and 6), and then display them in reverse order (from 9 to 0). While this would be much more efficient than the first embodiment, this embodiment does have the 5 drawback of consuming significant amounts of memory, if the GOP is large and/or if the resolution of the video is high.

Although various embodiments which incorporate the teachings of the present invention have been shown and described in detail herein, those skilled in the art can readily devise many other varied embodiments that still incorporate these teachings.

-11-

What is claimed is:

1. Apparatus comprising:

an encoder (102) for predictively encoding a video sequence to produce anchor frames and predicted frames of compressed information within a bitstream; and
5 an auxiliary file generator (104), coupled to said encoder, for producing an auxiliary file containing information pertaining to the location of anchor frames within the bitstream.

2. The apparatus of claim 1 further comprising:

10 a decoder (504) for decoding the compressed information to form a decoded video sequence;
an auxiliary file processor (506), coupled to said decoder, for using the auxiliary file to identify an anchor frame location within said bitstream that occurs in the bitstream prior to a selected frame and for causing the decoder (504) to decode the bitstream
15 starting at the identified anchor frame location.

3. The apparatus of claim 2 wherein said decoder (504) displays only the particular frame and blocks the display of all other decoded frames.

20 4. A method of encoding a video sequence comprising the steps of:

predictively encoding the video sequence to produce anchor frames and predicted frames in a bitstream; and
generating an auxiliary file comprising information pertaining to the location of the anchor frames within the bitstream.

-12-

5. A method of decoding a bitstream comprising compressed information containing anchor frames and predicted frames and an auxiliary file containing information pertaining to the location of anchor frames within the bitstream, the method comprising the steps of:

selecting a frame to be decoded;

5 identifying, using the auxiliary file, a location within the bitstream of an anchor frame prior to the selected frame; and

decoding the bitstream starting at the identified location.

6. The method of claim 5 further comprising the step of:

10 blocking the display of decoded frames until the selected frame is decoded; and

displaying the selected frame.

7. The method of claim 6 further comprising the step of repeatedly selecting frames that are spaced apart from one another within the bitstream to produce a fast forward display

15 effect.

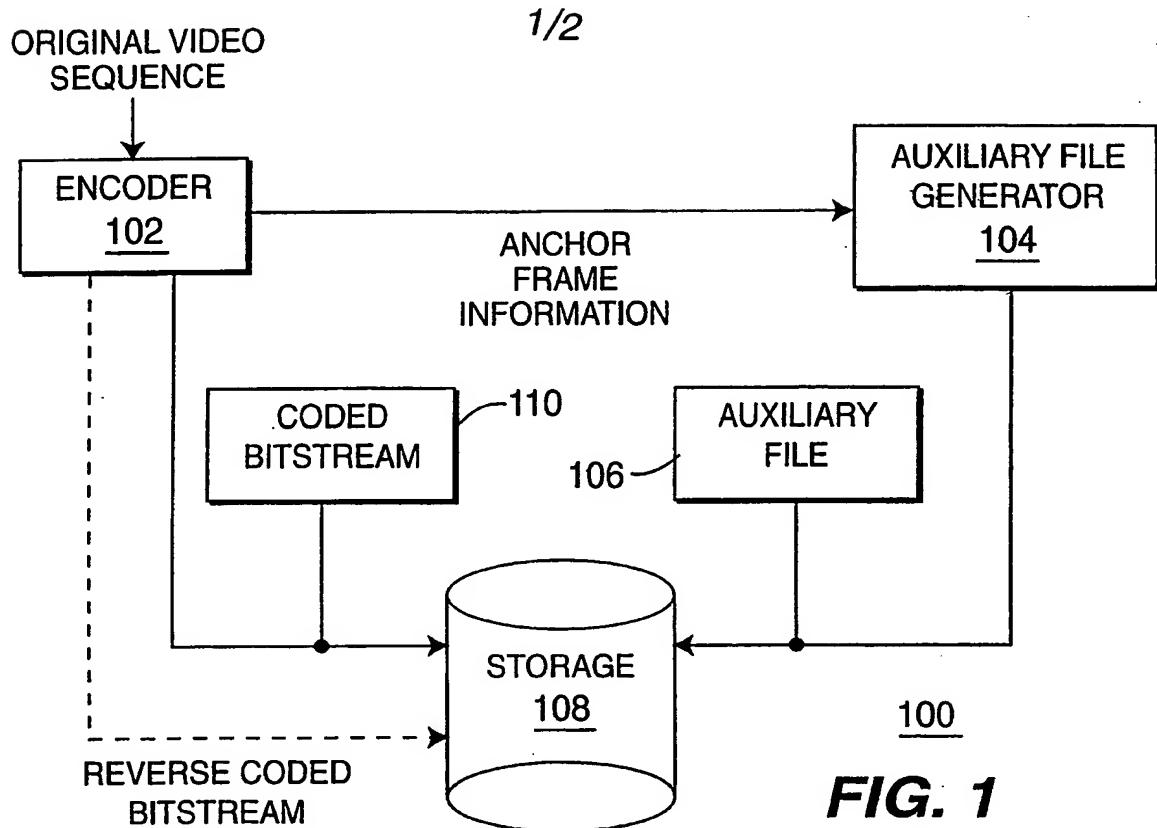
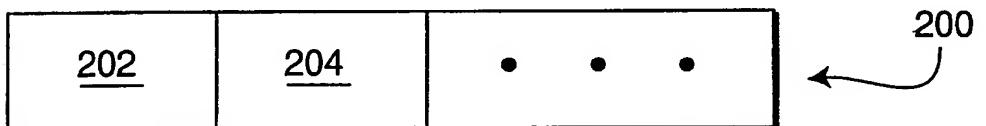
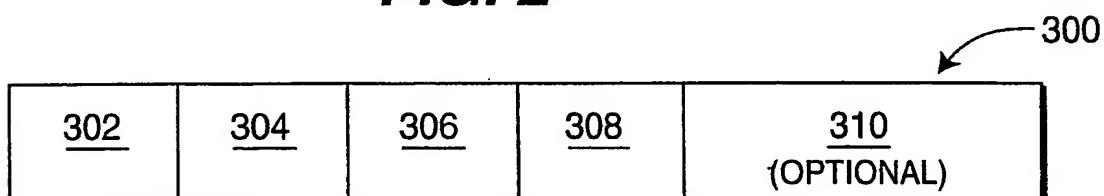
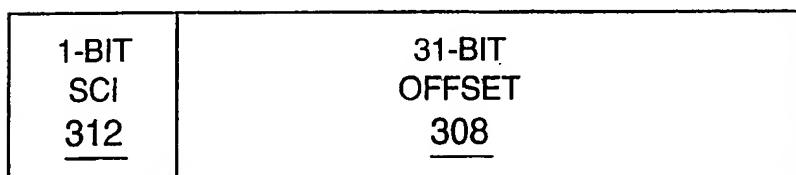
8. The method of claim 6 further comprising a step of repeatedly selecting frames that are spaced apart from one another, in reverse order within the bitstream to produce a reverse display effect.

20

9. The method of claim 6 further comprising the step of caching all frames that are decoded between two anchor frames.

10. The method of claim 6 wherein the selected frame is in a different scene.

25

**FIG. 1****FIG. 2****FIG. 3****FIG. 4**

2/2

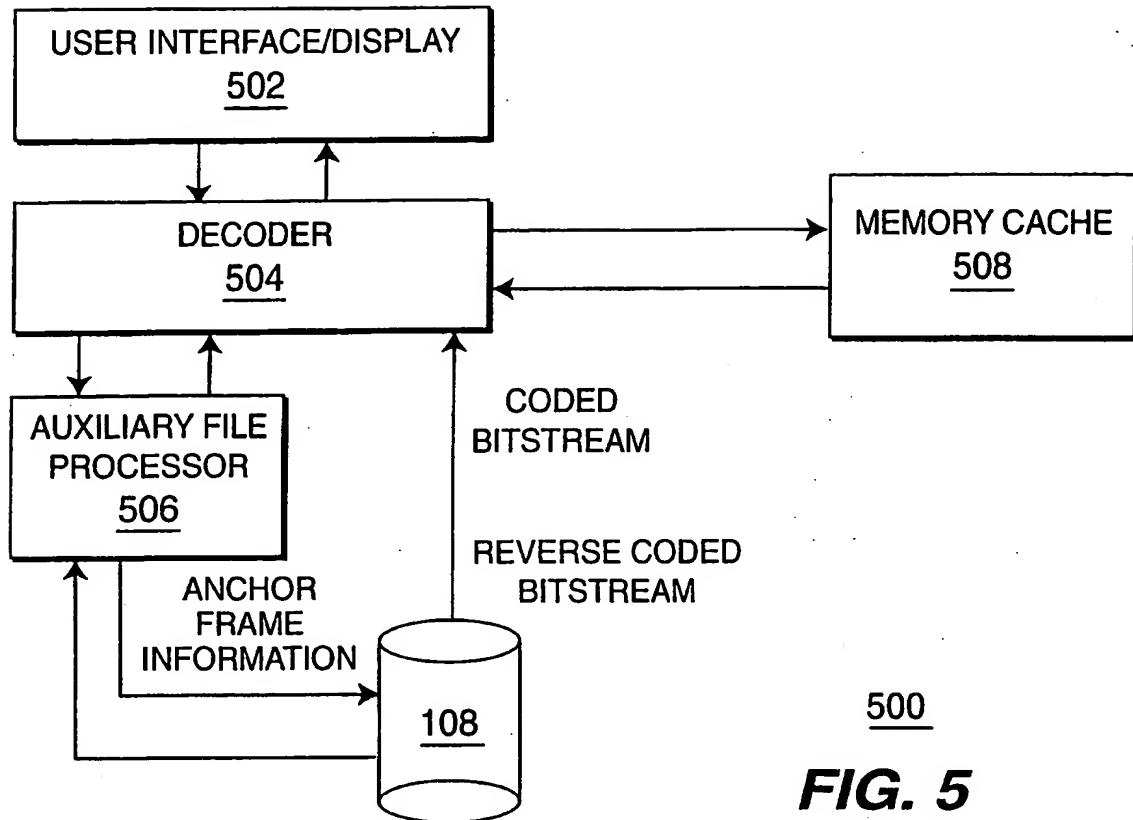


FIG. 5

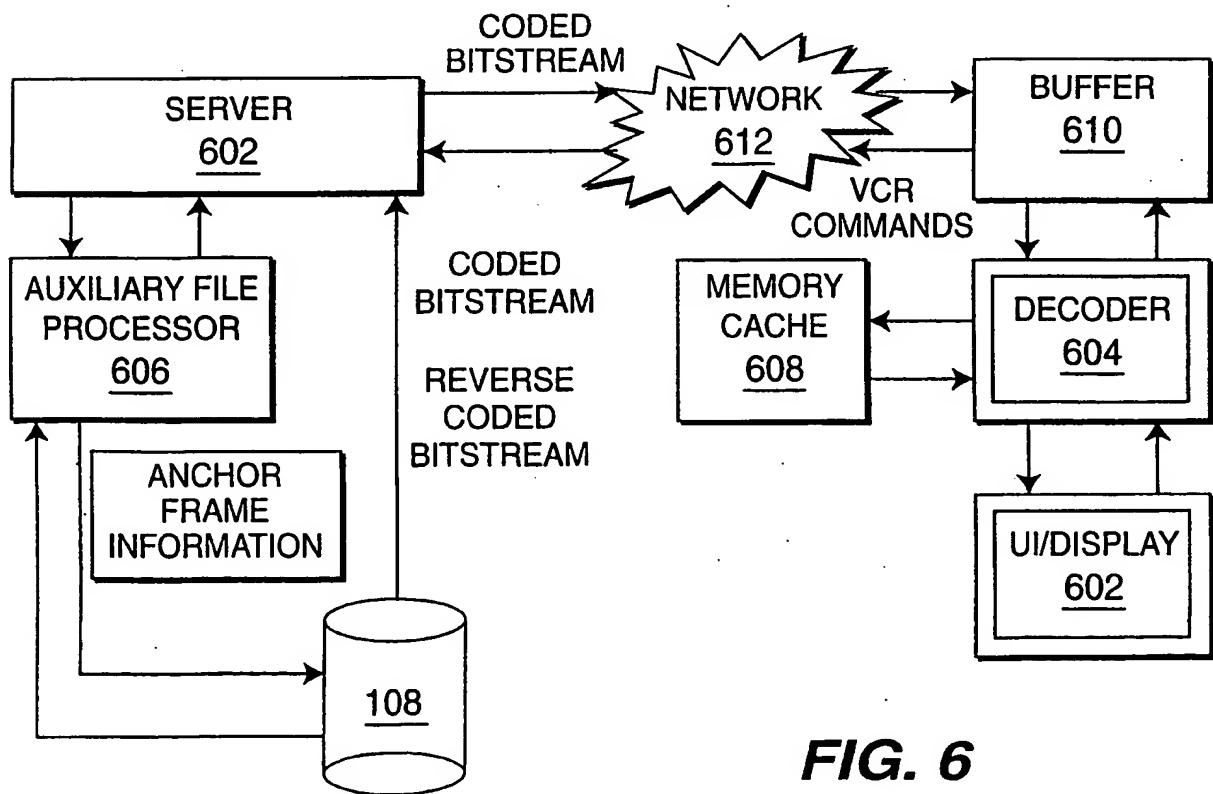


FIG. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/23375

A. CLASSIFICATION OF SUBJECT MATTER IPC 7 H04N5/783		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 7 H04N		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 725 399 A (SONY CORPORATION) 7 August 1996 (1996-08-07) the whole document ----	1-8, 10
X	EP 0 717 411 A (SONY CORPORATION) 19 June 1996 (1996-06-19) the whole document	1, 4
A	-----	2, 3, 5-8, 10
X	EP 0 729 153 A (HITACHI, LTD) 28 August 1996 (1996-08-28) the whole document	1, 4
A	-----	2, 3, 5-8, 10
<input type="checkbox"/> Further documents are listed in the continuation of box C.		<input checked="" type="checkbox"/> Patent family members are listed in annex.
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority, claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "S" document member of the same patent family		
Date of the actual completion of the international search	Date of mailing of the international search report	
11 February 2000	17/02/2000	
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl Fax: (+31-70) 340-3016	Authorized officer Verleye, J	

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 99/23375

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP 725399	A 07-08-1996	JP 8214264 A		20-08-1996
		AU 699801 B		17-12-1998
		AU 4226596 A		08-08-1996
		BR 9600279 A		23-12-1997
		CA 2167985 A		01-08-1996
		CN 1136260 A		20-11-1996
		US 5768466 A		16-06-1998
		US 5802242 A		01-09-1998
EP 717411	A 19-06-1996	JP 8168042 A		25-06-1996
		BR 9505921 A		23-12-1997
		CA 2165275 A		16-06-1996
		US 5719982 A		17-02-1998
		US 5923811 A		13-07-1999
EP 729153	A 28-08-1996	JP 8235833 A		13-09-1996
		CN 1135072 A		06-11-1996
		CN 1229235 A		22-09-1999
		EP 0930618 A		21-07-1999
		US 6002834 A		14-12-1999
		US 6009237 A		28-12-1999

THIS PAGE BLANK (USPTO)